

### 03 Day of the Week

In this program you will prompt the user to enter the day, month, and year. Your program will print out the day of the week for that date. Here is a sample output for the program:

```
Enter year: 1960
```

```
Enter month: 12
```

```
Enter day: 12
```

```
The day is Monday.
```

Assume that all the input values will be positive integers. The day will be in the range 1 through 31. The month will be in the range 1 through 12 where 1 is for January and 12 is for December. The year will be a four digit number in the range 1900 through 2100.

Your program will check the following:

- The year is between 1900 and 2100.
- The month is in the range 1 and 12.
- The day is in the range appropriate for that month. For months other than February, the range of days will be from 1 to 30 or 31 days. The range of days for February is between 1 and 28 or 29 depending on whether the year is a leap year or not.

All of the ranges are inclusive. You will first prompt the user to enter the year and then check whether the year is in the range specified. If it is not, then you will keep prompting the user to enter the correct year. After that you will prompt the user to enter the month and run a similar check. Lastly, you will prompt the user to enter the day and run a check. For example, the input 29, 2, 2012 is valid, but the input 29, 2, 1900 is not, because 2012 is a leap year and 1900 is not a leap year.

This algorithm was developed by Rev. Zeller. Let us define the quantities  $a$ ,  $b$ ,  $c$ , and  $d$  as follows:

- $a \equiv$  the month of the year, with March = 1, April = 2, ... with January = 11, and February = 12 of the preceding year
- $b \equiv$  the day of the month
- $c \equiv$  the year of the century
- $d \equiv$  the century, though not in the conventional sense. For example, after making adjustments for the months of January and February, for all years like 1910, 1949, 1984, the value of  $d = 19$ . And for years like 2003, 2010, 2025, the value of  $d = 20$ .

**Important:** In our calendar, the year begins in January and ends in December. In the calendar, used in the algorithm, the year begins in March and ends in February. Your program should internally make the adjustment and not expect the user to know this. For example, if in our calendar we have January 2009 (month = 1 and year = 2009), the program will make the adjustment so that month = 11 and year = 2008. If you do not make the adjustment you will not get the right result.

For example, 31 July 1929, gives  $a = 5$ ,  $b = 31$ ,  $c = 29$ , and  $d = 19$ . Similarly, 3 January 1988, gives  $a = 11$ ,  $b = 3$ ,  $c = 87$ , and  $d = 19$ .

Now compute the following quantities:

- $w = (13 * a - 1) // 5$
- $x = c // 4$
- $y = d // 4$
- $z = w + x + y + b + c - 2 * d$
- $r = z \% 7$
- $r = (r + 7) \% 7$  *[to take care of negative values of  $r$ ]*

$r$  gives the day of the week.  $r = 0$  represents Sunday,  $r = 1$  represents Monday, and so on. The output from your file should match the following statement:

The day is Monday.

Where *Monday* may be replaced by any day of the week.